UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/711,929	10/13/2004	Rajnish K. Chitkara	SYB/0110.01	5928

31779 7590 06/13/2008

JOHN A. SMART
201 LOS GATOS
SARATOGA RD, #161
LOS GATOS, CA 95030-5308

EXAMINER	
GORTAYO, DANGELINO N	
ART UNIT	PAPER NUMBER
2168	

MAIL DATE	DELIVERY MODE
06/13/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/711,929	CHITKARA ET AL.	
	Examiner	Art Unit	
	DANGELINO N. GORTAYO	2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 October 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-99* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-99* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *13 October 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *8/8/05*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-99 are pending.

Information Disclosure Statement

2. An initialed and dated copy of Applicant's IDS form 1449, filed 8/8/2005, is attached to the instant Office action.

Claim Rejections - 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. Claims 37-70 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claims are directed to a system claim but lacks any tangible hardware, memory, processors, input/outputs, and sources designed to provide automated encryption support for column data in a database. The claims are directed to software per se and are non-statutory. Proper correction is required.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-99 are rejected under 35 U.S.C. 103(a) as being anticipated by Newman et a. US Patent 7,266,699 B2) in view of Lei et al. (US Publication 2004/0255133 A1)

As per claim 1, Newman teaches "In a database system, a method for providing automated encryption support for column data," (see Abstract and column 1 lines 46-62)

"the method comprising: defining Structured Query Language (SQL) extensions for creating and managing column encryption keys, and for creating and managing database tables with encrypted column data;" (column 2 lines 20-27, column 4 lines 28-44, column 4 line 57 – column 5 line 12, column 5 lines 46-54, wherein a key management system which utilizes SQL as the standard query language provides encryption key management)

"receiving an SQL statement specifying creation of a particular column encryption key;" (column 2 lines 28-40, column 7 lines 45-52, wherein a command to encrypt a column causes a key to be created)

"and in response to a subsequent database operation that requires the particular column data that has been encrypted, automatically decrypting the particular column data for use by the database operation." (column 5 lines 37-54, column 8 lines 30-63, wherein encrypted data is automatically decrypted in response to an authorized user accessing the encrypted column data)

While Newman teaches that encrypted database tables are able to be viewed and processed by authorized users (column 2 lines 41-57, column 7 lines 16-25, column 7 lines 45-56), Newman does not specifically teach "receiving an SQL statement

specifying creation of a database table having particular column data encrypted with said particular column encryption key;"

Lei teaches "receiving an SQL statement specifying creation of a database table having particular column data encrypted with said particular column encryption key;" (paragraphs 0019, 0020, 0066, 0067 wherein a column is selected to be encrypted and stored in the database, to create encrypted data tables based on keys)

It would have been obvious for one of ordinary skill in the art to combine Newman's method of providing a transparent encryption infrastructure for databases with Lei's method of storing and updating encrypted tables. This gives the user the ability to save encrypted data in the database. The motivation for doing so would be to more efficiently provide transparent access to user applications accessing sensitive data protected by encryption (paragraphs 0007, 0008)

As per claim 2, Newman teaches "columns that are not specified to be encrypted are stored in unencrypted format, for minimizing encryption overhead." (column 7 lines 17-24)

As per claim 3, Newman teaches "the automated encryption support operates as an internal built-in feature of the database system, without use of an add-on library." (column 3 lines 52-62)

As per claim 4, Newman teaches the SQL statement specifying creation of a particular encryption key is received from a user serving as a system security officer. (column 10 lines 23-31)

As per claim 5, Lei teaches the SQL statement specifying creation of a database table may be received from a user other than the system security officer. (paragraph 0035)

As per claim 6, Newman teaches the SQL statement specifying creation of a particular encryption key comprises a CREATE ENCRYPTION KEY command. (column 7 lines 45-52)

As per claim 7, Newman teaches the CREATE ENCRYPTION KEY command includes:

CREATE ENCRYPTION KEY keyname
[AS DEFAULT] [FOR algorithm]
[WITH [KEYLENGTH keysize]
[PASSWD passphrase]
[INIT_VECTOR [RANDOM | NULL]]
[PAD [RANDOM | NULL]]]

as its syntax. (column 8 lines 5-22)

As per claim 8, Lei teaches the SQL statement specifying creation of a database table having particular column data encrypted comprises a CREATE TABLE command that allows specification of one or more columns to be encrypted. (paragraph 0066)

As per claim 9, Lei teaches the CREATE TABLE command includes:

CREATE TABLE tablename
(colname1 datatype [encrypt [with [db.[owner].]keyname],
colname2 datatype [encrypt [with [db.[owner].]keyname])

as its syntax. (paragraph 0066)

As per claim 10, Lei teaches receiving an SQL statement specifying alteration of a previously-created database table so as to encrypt particular column data. (paragraph 0061, 0062, 0063)

As per claim 11, Lei teaches the SQL statement specifying alteration of a previously created database table comprises an ALTER TABLE command. (paragraph 0061, 0062, 0063)

As per claim 12, Lei teaches the ALTER TABLE command includes:

ALTER TABLE tablename MODIFY column_name
[[datatype] [null|not null]]
[decrypt | encrypt [with [db.[owner].]keyname]]

as its syntax. (paragraph 0061, 0062, 0063)

As per claim 13, Newman teaches the encryption support works transparently with existing database applications.(column 4 lines 57-64)

As per claim 14, Newman teaches the database system includes a database server and one or more database clients, and wherein method steps implementing the encryption support are embodied at the database server. (column 4 lines 38-48)

As per claim 15, Newman teaches the database system includes a back-end server tier and a middleware tier, and wherein method steps implementing the encryption support are embodied at the back-end server tier. (column 4 lines 38-48)

As per claim 16, Newman teaches after creation of the particular column encryption key, protecting the particular column encryption key with a user-supplied password. (column 2 lines 52-59, column 6 lines 11-21)

As per claim 17, Newman teaches the user-supplied password must be supplied before the system allows use of the particular column encryption key for database operations. (column 6 lines 11-21)

As per claim 18, Newman teaches the user-supplied password is supplied using a SET ENCRYPTION PASSWD command. (column 10 lines 23-62)

As per claim 19, Newman teaches the SET ENCRYPTION PASSWD command includes:

SET ENCRYPTION PASSWD password FOR keyname

as its syntax. (column 10 lines 23-62)

As per claim 20, Newman teaches a user seeking to decrypt column data must supply said user-supplied password and must have necessary database privileges before decrypting the column data with the particular column encryption key. (column 6 line 40 - column 7 line 5)

As per claim 21, Newman teaches the user-supplied password is supplied using a SET ENCRYPTION PASSWD command. (column 6 line 40 - column 7 line 5)

As per claim 22, Newman teaches providing a command to grant decryption permission to others. (column 10 lines 23-62)

As per claim 23, Newman teaches the command to grant decryption permission includes:

GRANT DECRYPT ON table.column TO user_or_role_list

as its syntax. (column 10 lines 23-62)

As per claim 24, Newman teaches the database system internally stores in encrypted format any column encryption keys that have been created. Column 5 lines 12-37)

As per claim 25, Newman teaches the database system stores encrypted column data internally as variable binary (VARBINARY) data. (column 10 lines 19-23)

As per claim 26, Newman teaches the database system presents users a user-defined field type for column data that has been encrypted, even though the column data is stored internally as variable binary data. (column 7 lines 16-24)

As per claim 27, Newman teaches the database system preserves any user-defined data type for the particular column data so that the database system employs a correct data type for processing queries and returning query results. (column 7 line 64 – column 8 line 3)

As per claim 28, Newman teaches the database system stores the user-defined data type for the particular column data in a system catalog of the database system. (column 7 line 64 – column 8 line 3, column 8 line 30-35)

As per claim 29, Newman teaches the particular column encryption key created comprises a symmetric encryption key.(column 2 lines 41-52)

As per claim 30, Newman teaches a single column encryption key is used for each column to be encrypted. (column 4 lines 61-64)

As per claim 31, Newman teaches a single column encryption key may be shared by multiple columns to be encrypted. (column 5 lines 13-30)

As per claim 32, Newman teaches the particular column encryption key is itself encrypted to a key-encrypting key constructed from a user-supplied password. (column 5 lines 13-30)

As per claim 33, Newman teaches the particular column encryption key is itself stored on disk in encrypted format using Advanced Encryption Standard (AES) encryption. (column 4 lines 19-28, column 5 lines 31-37)

As per claim 34, Newman teaches the user-supplied password may comprise a hex literal. (column 10 lines 8-18)

As per claim 35, Newman teaches the user-supplied password is itself transformed into a symmetric encryption key, using a random salt, internal static data, and SHA-1 hashing algorithm. (column 4 lines 19-28, column 10 lines 30-62)

As per claim 36, Newman teaches said Structured Query Language (SQL) extensions for creating and managing column encryption keys include a clause for instructing the database system to create a default key for encrypting columns. (column 6 lines 11-22)

As per claim 37, Newman teaches A database system providing automated encryption support for column data," (see Abstract and column 1 lines 46-62)

"the system comprising: a parser that supports Structured Query Language (SQL) extensions for creating and managing column encryption keys, and for creating and managing database tables with encrypted column data;" (column 2 lines 20-27, column 4 lines 28-44, column 4 line 57 – column 5 line 12, column 5 lines 46-54, wherein a key management system which utilizes SQL as the standard query language provides encryption key management)

"and an execution unit, operating in response to SQL statements parsed by the parser, for creating a particular column encryption key," (column 2 lines 28-40, column 7 lines 45-52, wherein a command to encrypt a column causes a key to be created)

"and for automatically decrypting the particular column data for use by a subsequent database operation that requires the particular column data that has been encrypted." (column 5 lines 37-54, column 8 lines 30-63, wherein encrypted data is automatically decrypted in response to an authorized user accessing the encrypted column data)

While Newman teaches that encrypted database tables are able to be viewed and processed by authorized users (column 2 lines 41-57, column 7 lines 16-25, column 7 lines 45-56), Newman does not specifically teach "for creating a database table having particular column data encrypted with said particular column encryption key,"

Lei teaches "for creating a database table having particular column data encrypted with said particular column encryption key,"

(paragraphs 0019, 0020, 0066, 0067 wherein a column is selected to be encrypted and stored in the database, to create encrypted data tables based on keys)

It would have been obvious for one of ordinary skill in the art to combine Newman's method of providing a transparent encryption infrastructure for databases with Lei's method of storing and updating encrypted tables. This gives the user the ability to save encrypted data in the database. The motivation for doing so would be to more efficiently provide transparent access to user applications accessing sensitive data protected by encryption (paragraphs 0007, 0008)

As per claim 38, Newman teaches columns that are not specified to be encrypted are stored in unencrypted format, for minimizing encryption overhead. (column 7 lines 17-24)

As per claim 39, Newman teaches the automated encryption support operates as an internal built-in feature of the database system, without use of an add-on library. (column 3 lines 52-62)

As per claim 40, Newman teaches the SQL statement specifying creation of a particular encryption key is received from a user serving as a system security officer. (column 10 lines 23-31)

As per claim 41, Lei teaches the SQL statement specifying creation of a database table may be received from a user other than the system security officer. (paragraph 0035)

As per claim 42, Newman teaches the SQL statement specifying creation of a particular encryption key comprises a CREATE ENCRYPTION KEY command. (column 7 lines 45-52)

As per claim 43, Newman teaches the CREATE ENCRYPTION KEY command includes:

```
CREATE ENCRYPTION KEY keyname
[AS DEFAULT] [FOR algorithm]
[WITH [KEYLENGTH keysize]
[PASSWD passphrase]
[INIT_VECTOR [RANDOM | NULL]]
[PAD [RANDOM | NULL]]]
```

as its syntax. (column 8 lines 5-22)

As per claim 44, Lei teaches the SQL statement specifying creation of a database table having particular column data encrypted comprises a CREATE TABLE command that allows specification of one or more columns to be encrypted. (paragraph 0066)

As per claim 45, Lei teaches the CREATE TABLE command includes:

CREATE TABLE tablename
(colname1 datatype [encrypt [with [db.[owner].]keyname],
colname2 datatype [encrypt [with [db.[owner].]keyname])

as its syntax. (paragraph 0066)

As per claim 46, Lei teaches a module for receiving an SQL statement specifying alteration of a previously created database table so as to encrypt particular column data. (paragraph 0061, 0062, 0063)

As per claim 47, Lei teaches the SQL statement specifying alteration of a previously created database table comprises an ALTER TABLE command. (paragraph 0061, 0062, 0063)

As per claim 48, Lei teaches the ALTER TABLE command includes:

ALTER TABLE tablename MODIFY column_name
[[datatype] [null|not null]]
[decrypt | encrypt [with [db.[owner].]keyname]]

as its syntax. (paragraph 0061, 0062, 0063)

As per claim 49, Newman teaches the encryption support works transparently with existing database applications. (column 4 lines 57-64)

As per claim 50, Newman teaches the database system includes a database server and one or more database clients, and wherein the encryption support is provided by the database server. (column 4 lines 38-48)

As per claim 51, Newman teaches the database system includes a back-end server tier and a middleware tier, and wherein the encryption support is provided by the back-end server tier. (column 4 lines 38-48)

As per claim 52, Newman teaches the system protects the particular column encryption key with a user-supplied password. (column 2 lines 52-59, column 6 lines 11-21)

As per claim 53, Newman teaches the user-supplied password must be supplied before the system allows use of the particular column encryption key for database operations. (column 6 lines 11-21)

As per claim 54, Newman teaches the user-supplied password is supplied using a SET ENCRYPTION PASSWD command. (column 10 lines 23-62)

As per claim 55, Newman teaches the SET ENCRYPTION PASSWD command includes:

SET ENCRYPTION PASSWD password FOR keyname

as its syntax. (column 10 lines 23-62)

As per claim 56, Newman teaches a user seeking to decrypt column data must supply said user-supplied password and must have necessary database privileges before decrypting the column data with the particular column encryption key. (column 6 line 40 - column 7 line 5)

As per claim 57, Newman teaches providing a command to grant decryption permission to others. (column 10 lines 23-62)

As per claim 58, Newman teaches the command to grant decryption permission includes:

GRANT DECRYPT ON table.column TO user_or_role_list

as its syntax. (column 10 lines 23-62)

As per claim 59, Newman teaches the database system internally stores in encrypted format any column encryption keys that have been created. (Column 5 lines 12-37)

As per claim 60, Newman teaches the database system stores encrypted column data internally as variable binary (VARBINARY) data. (column 10 lines 19-23)

As per claim 61, Newman teaches the database system presents users a user-defined field type for column data that has been encrypted, even though the column data is stored internally as variable binary data. (column 7 lines 16-24)

As per claim 62, Newman teaches the database system preserves any user-defined data type for the particular column data so that the database system employs a correct data type for processing queries and returning query results. (column 7 line 64 – column 8 line 3)

As per claim 63, Newman teaches the database system stores the user-defined data type for the particular column data in a system catalog of the database system. (column 7 line 64 – column 8 line 3, column 8 line 30-35)

As per claim 64, Newman teaches the particular column encryption key created comprises a symmetric encryption key. (column 2 lines 41-52)

As per claim 65, Newman teaches a single column encryption key is used for each column to be encrypted. (column 4 lines 61-64)

As per claim 66, Newman teaches the particular column encryption key is itself encrypted to a key-encrypting key constructed from a user-supplied password. (column 5 lines 13-30)

As per claim 67, Newman teaches the particular column encryption key is itself stored on disk in encrypted format using Advanced Encryption Standard (AES) encryption. (column 4 lines 19-28, column 5 lines 31-37)

As per claim 68, Newman teaches the user-supplied password may comprise a hex literal. (column 10 lines 8-18)

As per claim 69, Newman teaches the user-supplied password is itself transformed into a symmetric encryption key, using a random salt, static internal data and SHA-1 hashing algorithm. (column 4 lines 19-28, column 10 lines 30-62)

As per claim 70, Newman teaches said Structured Query Language (SQL) extensions for creating and managing column encryption keys include a clause for instructing the database system to create a default key for encrypting columns. (column 6 lines 11-22)

As per claim 71, Newman teaches "In a database system, a method for encrypting column data," (see Abstract and column 1 lines 46-62)

"the method comprising: in response to a first query language statement, creating an encryption key for encrypting a particular column of a database table;" (column 2 lines 28-40, column 7 lines 45-52, wherein a command to encrypt a column causes a key to be created)

"and during a subsequent database operation requiring column data from the particular column, automatically decrypting the column data for use by the database operation." (column 5 lines 37-54, column 8 lines 30-63, wherein encrypted data is automatically decrypted in response to an authorized user accessing the encrypted column data)

While Newman teaches that encrypted database tables and columns are able to be viewed and processed by authorized users (column 2 lines 41-57, column 7 lines 16-25, column 7 lines 45-56), Newman does not specifically teach "in response to a second query language statement, encrypting the particular column using said encryption key;"

Lei teaches "in response to a second query language statement, encrypting the particular column using said encryption key;" (paragraphs 0019, 0020, 0066, 0067 wherein a column is selected to be encrypted and stored in the database, to create encrypted data tables based on keys)

It would have been obvious for one of ordinary skill in the art to combine Newman's method of providing a transparent encryption infrastructure for databases with Lei's method of storing and updating encrypted tables. This gives the user the ability to save encrypted data in the database. The motivation for doing so would be to

more efficiently provide transparent access to user applications accessing sensitive data protected by encryption (paragraphs 0007, 0008)

As per claim 72, Newman teaches assigning privileges to users for creating an encryption key for encrypting column data. (column 5 lines 3-12, column 10 lines 23-31)

As per claim 73, Newman teaches in response to a request to create an encryption key from a particular user, determining whether the particular user has sufficient privileges to create an encryption key. (column 5 lines 3-12, lines 38-54)

As per claim 74, Newman teaches the encryption key is itself encrypted to a key-encrypting key constructed from a user-supplied password. (column 5 lines 13-30)

As per claim 75, Newman teaches the encryption key is encrypted using Advanced Encryption Standard (AES) encryption. (column 4 lines 19-28, column 5 lines 31-37)

As per claim 76, Newman teaches the user-supplied password may comprise a hex literal. (column 10 lines 8-18)

As per claim 77, Newman teaches the user-supplied password is itself transformed into a symmetric encryption key, using a random salt, static internal data and SHA-1 hashing algorithm. (column 4 lines 19-28, column 10 lines 30-62)

As per claim 78, Newman teaches the database system stores encrypted column data internally as variable binary (VARBINARY) data. (column 10 lines 19-23)

As per claim 79, Newman teaches columns of the database table that are not specified to be encrypted are stored in unencrypted format. (column 7 lines 17-24)

As per claim 80, Newman teaches the system implements said first and second statements as SQL extensions for creating and managing encryption keys and for creating and managing database tables with encrypted column data. (column 10 lines 23-31)

As per claim 81, Newman teaches said SQL extensions include a CREATE ENCRYPTION KEY command for creating an encryption key. (column 7 lines 45-52)

As per claim 82, Newman teaches said CREATE ENCRYPTION KEY command includes attributes specifying an encryption key name and a user-supplied password. (column 7 lines 45-52, column 8 lines 5-22)

As per claim 83, Lei teaches said SQL extensions include a CREATE TABLE command having an attribute that allows specification of at least one column to be encrypted. (paragraph 0066)

As per claim 84, Lei teaches said CREATE TABLE command syntax includes attributes specifying a table name, one or more columns to be encrypted, and an encryption key name. (paragraph 0066)

As per claim 85, Lei teaches said second query language statement includes a request specifying alteration of a previously-created table so as to encrypt particular column data. (paragraph 0061, 0062, 0063)

As per claim 86, Newman teaches a user subsequently requiring use of the encrypted column data must provide a user-supplied password for unlocking the encryption key for the particular column. (column 2 lines 52-59, column 6 lines 11-21)

As per claim 87, Newman teaches receiving an SQL statement specifying creation of a default key encryption password. (column 6 lines 11-22)

As per claim 88, Newman teaches the SQL statement specifying creation of a default key encryption password specifies a default password value that is encrypted by a system stored procedure, for storage in a system table of a particular database. (column 6 lines 11-22)

As per claim 89, Newman teaches receiving an SQL statement specifying creation of an encryption keypair. (column 2 lines 28-40)

As per claim 90, Newman teaches the SQL statement specifying creation of an encryption keypair comprises a CREATE ENCRYPTION KEYPAIR command. (column 10 lines 6-62)

As per claim 91, Newman teaches the CREATE ENCRYPTION KEYPAIR command includes:

CREATE ENCRYPTION KEYPAIR keypairname
[FOR algorithm]
[WITH [KEYLENGTH keysize]
[PASSWD passphrase | LOGIN_PASSWD]

as its syntax. (column 10 lines 6-62)

As per claim 92, Newman teaches receiving an SQL statement specifying alteration of a particular encryption key or keypair. (column 10 lines 52-62)

As per claim 93, Newman teaches receiving an SQL statement specifying dropping a particular encryption key or keypair. (column 10 lines 62-65)

As per claim 94, Newman teaches receiving an SQL statement granting rights to a particular encryption key or keypair. (column 10 lines 6-62)

As per claim 95, Newman teaches receiving an SQL statement revoking said rights that have been granted to a particular encryption key or keypair. (column 10 lines 62-65)

As per claim 96, Newman teaches the said rights granted for the particular encryption key or keypair comprise SELECT query execution rights, for selecting encrypted data. (column 10 line 65 – column 11 line 16)

As per claim 97, Newman teaches the said rights granted for the particular encryption key or keypair comprise ALTER query execution rights, for altering the encryption key or keypair. (column 10 line 65 – column 11 line 16)

As per claim 98, Newman teaches A computer-readable medium having processor-executable instructions for performing the method of claim 71. (column 4 lines 38-48)

As per claim 99, Newman teaches A downloadable set of processor-executable instructions for performing the method of claim 71. (column 4 lines 38-48)

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Wessman (US Patent 7,111,005 B1)

Any inquiry concerning this communication or earlier communications from the examiner should be directed to DANGELINO N. GORTAYO whose telephone number is (571)272-7204. The examiner can normally be reached on M-F 7:30-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim T. Vo can be reached on (571)272-3642. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Tim T. Vo/
Supervisory Patent Examiner, Art Unit 2168

Dangelino N. Gortayo
Examiner

Tim T. Vo
SPE